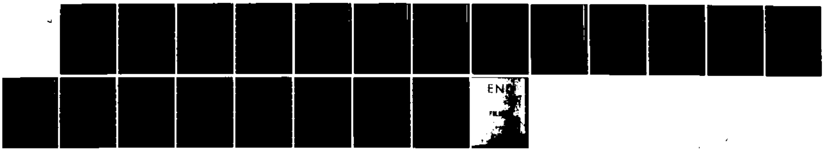
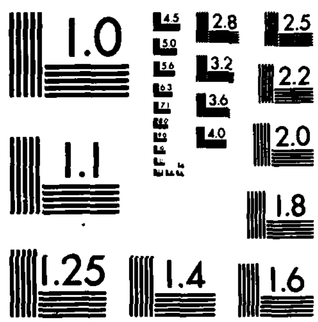
AD-A143 360 SENSOR NOISE AND KALMAN FILTER FOR AIDED INERTIAL NAVIGATION SYSTEM(U) SOUTHERN METHODIST UNIV DALLAS TEX DEPT OF ELECTRICAL ENGINEERING G S GREWAL MAY 84 1/1

UNCLASSIFIED AFOSR-TR-84-0556 AFOSR-82-0006 F/G 17/7 NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

Sponsored by the

AIR FORCE OFFICE OF SCIENTIFIC RESEARCH

FINAL REPORT

SENSOR NOISE AND KALMAN FILTER FOR AIDED INERTIAL NAVIGATION SYSTEM

Prepared by: Dr. Gurmohan S. Grewal

Academic Rank: Professor

Department and University: Electrical Engineering, Southern University

Contract No: AFOSR-82-0006

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

AD-A143360

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION: UNCLASSIFIED

1b. RESTRICTIVE MARKINGS:

2a. SECURITY CLASSIFICATION AUTHORITY:

2b. DECLASSIFICATION/DOWNGRADING SCHEDULE:

3. DISTRIBUTION/AVAILABILITY OF REPORT: Approved for public release; distribution unlimited.

4. PERFORMING ORGANIZATION REPORT NUMBER(S):

5. MONITORING ORGANIZATION REPORT NUMBER(S): AFOSR-TR- 84-0556

6a. NAME OF PERFORMING ORGANIZATION: Southern University

6b. OFFICE SYMBOL (If applicable):

7a. NAME OF MONITORING ORGANIZATION: Air Force Office of Scientific Research

6c. ADDRESS (City, State and ZIP Code): Electrical Engineering, Los Angeles, CA 90089

7b. ADDRESS (City, State and ZIP Code): Directorate of Mathematical & Information Sciences, Bolling AFB DC 20332

8a. NAME OF FUNDING/SPONSORING ORGANIZATION: AFOSR

8b. OFFICE SYMBOL (If applicable): NM

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER: AFOSR 82-0006

8c. ADDRESS (City, State and ZIP Code): Bolling AFB DC 20332

10. SOURCE OF FUNDING NOS.

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT NO.
61102F	2304	D9	

11. TITLE (Include Security Classification): SENSOR NOISE AND KALMAN FILTER FOR AIDED INTERTIAL NAVIGATION SYSTEM

12. PERSONAL AUTHOR(S): Gurmohan S. Grewal

13a. TYPE OF REPORT: Final

13b. TIME COVERED: FROM 15 MAR 82 TO 14 MAR 83

14. DATE OF REPORT (Yr., Mo., Day): May 84

15. PAGE COUNT: 19

16. SUPPLEMENTARY NOTATION:

17. COSATI CODES

FIELD	GROUP	SUB. GR.

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number):

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Inertial Navigation System, barometric altimeter, TACAN, and ILS are used to achieve a synergistic combination of the outputs of individual subsystems. Kalman filter is used to provide an ideal method for data processing in ths multisensor navigation system. The filter design begins with the development of mathematical and statistical error models to describe the truth system. The truth model is simplified and reduced, in steps, to lower the computation burden on the on-board computer. The covariance analysis and the Monte Carlo methods of testing the performance of the Kalman filters based on reduced and simplified system models are discussed. Suggestions for further research in the area of fault detection and isolation are offered.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT: UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS ☐

21. ABSTRACT SECURITY CLASSIFICATION: UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL: BRIAN W. WOODRUFF

22b. TELEPHONE NUMBER (Include Area Code): (202) 767-5025

22c. OFFICE SYMBOL: NM

DD FORM 1473, 83 APR — EDITION OF 1 JAN 73 IS OBSOLETE.

84 07 24 046

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

SENSOR NOISE AND KALMAN FILTER FOR AIDED INERTIAL NAVIGATION SYSTEM

by

Gurmohan S. Grewal

ABSTRACT

Inertial Navigation System, barometric altimeter, TACAN, and ILS are used to achieve a synergistic combination of the outputs of individual subsystems. Kalman filter is used to provide an ideal method for data processing in this multisensor navigation system. The filter design begins with the development of mathematical and statistical error models to describe the truth system. The truth model is simplified and reduced, in steps, to lower the computation burden on the on-board computer. The covariance analysis and the Monte Carlo methods of testing the performance of the Kalman filters based on reduced and simplified system models are discussed. Suggestions for further research in the area of fault detection and isolation are offered.

A-1

Acknowledgement

The author would like to thank the Air Force Systems Command, the Air Force Office of Scientific Research, and the Southeastern Center for Electrical Engineering Education for providing him with the opportunity to spend a very worthwhile and interesting summer at the Air Force Avionics Laboratory, Wright-Patterson AFB, Ohio.

Finally, he would like to thank Capt Nick Drobot for suggesting this area of research and for his assistance and guidance.

I. INTRODUCTION

Inertial Navigation Systems (INS) and non-inertial navigation aids such as TACAN, ILS, Loran, OMEGA, navigation satellites, etc., have been used in a number of multisensor-based navigation systems. Outputs of the individual subsystems are combined synergistically. The software accomplishes this combination ideally utilizing the data from the subsystems to yield much more accurate results than these subsystems can provide unaided. Data processing algorithm, called Kalman filter, provides a systemic and logical method of weighing various sources of information to produce a best estimate of the quantities of interest.

This research is concerned with the development of a Kalman filter that combines the data from a baro-inertial navigation with the range and bearing measurements of a TACAN system during the cruise portion of the flight and the measurements from an ILS system during the descent and the final approach phase of a flight (Ref 1). The resulting filter provides the position and the attitude of the aircraft.

The performance of a Kalman filter is dependent upon adequate mathematical and statistical models to describe the true system including system and measurement dynamics, system disturbances and measurement errors, and initial condition information. These models are formulated in the state space. There are two approaches available for the state space formulation of the models: the "total" state space models and the "error" state space models. In the total state space formulation, position, velocity, and attitude are among the state variables, and the measurements include accelerometer outputs and signals from TACAN or ILS. The resulting vehicle dynamics equations are nonlinear, high frequency, and are not adequately developed for use in the Kalman filter. In the error state space formulation, the errors in the inertial navigation system position, velocity, and attitude values are among the state variables, and measurements are composed of the differences between the inertial and the external source data. The resultant vehicle dynamics equations for the error state space formulation are low frequency, linear, and fairly well developed for use in the Kalman filter (Ref 2, 3, 4). Consequently, the error state space formulation, which is also called "indirect" filter, is adopted. Further, the indirect filter can be implemented in two ways: feedforward and feedback. In the case of feedforward, the output of the filter, which is the optimal estimates of the errors between the inertial system outputs and the true values, is subtracted from the inertial system outputs to obtain the best available estimate of the vehicle position, velocity, and attitude. The inertial system is unaware of the existence of the Kalman filter. The inertial system is free to drift with unbounded errors. As these errors get large, the adopted model of the inertial system becomes invalid, resulting in filter "divergence." On the other hand, in the feedback configuration, the output of the filter, which is the error between the true values and the inertial values, is fed back to the inertial system to obtain a set of corrected inertial outputs. Thus the inertial system errors are not allowed to grow unbounded. Moreover, the filter need not propagate the estimates of the error state variables. Hence, the feedback configuration is preferred.

If the comprehensive truth models are used in the development of Kalman filter, the resulting filter will require extensive memory and computation time, making it impractical for the limited on-board computer to handle the problem. The computation load is approximately proportional to the third power of the number of states required for modelling the system dynamics. Therefore, simplified models, rather than the truth models, are used in the filter development. The models simplification will result in performance degradation. In order to make intelligent approximations and assumptions necessary to obtain workable models, it is important to thoroughly understand the laws governing the involved system. Resulting performance degradation can be analyzed using covariance analysis and the Monte Carlo methods. The Air Force has fully developed, unclassified, transportable software packages for vehicle trajectory generation (Ref 5), covariance analysis (Ref 6), Monte Carlo analysis (Ref 7), and for plotting the results of Monte Carlo analysis (Ref 8), specifically to aid the testing and evaluation of the Kalman filter.

II. OBJECTIVES

The effort involved for this Summer Faculty Program had two main objectives:

(1) To provide an option for the injection of random errors into the outputs of the INS, TACAN, and air data sensor models of the Digital Avionics Information System (DAIS). Each of these random error sources is to be capable of interruption by the setting of an appropriate flag bit in a control word.

(2) After a detailed study of the available comprehensive TACAN/ILS-aided baro-inertial navigation system models (Ref 2, 3, 4, 9), develop reduced states simplified models in order to obtain a workable Kalman filter. The simplifications and the reductions are to be implemented in steps in order that the Kalman filter resulting from each approximation can be tested for the performance sensitivity to that particular approximation. There will not be enough time to complete the performance analysis, but the strategy for the step-by-step models simplification is to be well established by the end of this ten-week summer period.

III. RANDOM ERRORS INJECTION INTO DAIS SENSOR MODELS

In order to simulate various error conditions, a control word is defined with the following bit assignments:

Control Word Bit Assignments

Error Model	Signal	Bit No 1=ON, 0=OF	Default Value
INS	Actual Latitude	1	1
	1 Lat. Error	2	0
	5 Lat. Error	3	0
	Actual Longitude	4	1
	1 Long. Error	5	0
	5 Long. Error	6	0
TACAN	Actual Range	7	1
	1 Range Error	8	0
	5 Range Error	9	0
	Actual Bearing	10	1
	1 Bearing Error	11	0
	5 Bearing Error	12	0
Air Data	Actual TAS	13	1
	1 TAS Error	14	0
	5 TAS Error	15	0
	Not Used	16	0

The DEC-10 FORTRAN statement for the control word is inserted in the SCEN subroutine of DAIS:

```
IERROR = "0XXXXX"
           |||||
           ||||└──── INS Latitude Error
           |||└───── INS Longitude Error
           ||└────── TACAN Range Error
           |└─────── TACAN Bearing Error
           └──────── Air Data Error
```

Default value IERROR = "011111"

Octal Digit Values =	1 ,	NO Noise
	2 ,	1 Noise
	4 ,	5 Noise

The control word is set by the operator at the beginning of the simulation run.

3.1 Random Noise Model

A normally distributed random variable N of zero mean and unit variance is obtained from two independent samples U_1 and U_2 of uniform distribution between zero and one by the following equation:

$$N = \sqrt{(-2 \ln u_1)} \sin 2\pi u_2$$

A second normally distributed is obtained from the above equation by changing the SIN to COS:

$$N = \sqrt{-2 \ln u_1} \cos 2\pi w_2$$

The time correlation is introduced as follows:

$$x_n = \exp(-\Delta t/\tau)\, x_{n-1} + \sqrt{(1 - \exp(-2\Delta t/\tau))}\, N_n$$

where

n = nth iteration
Δt = iteration interval
τ = correlation time

The error e_n of mean M and standard deviation   is obtained by the following equation:

$$e_n = \sigma x_n + M$$

Since this random noise is used repeatedly, a subroutine called RNOISE is generated for it. FORTRAN coding for this subroutine is as follows:

```
      SUBROUTINE RNOISE(XLAST,DELT,TAU,SIGMA,ERR,XCUR)
      DOUBLE PRECISION X,X1,X2,RHO,ERR,XCUR,XLAST,PI2
      IF(X2.EQ.0.) GO TO 5
      X=X2
      X2=0.
      GO TO 10
    5 CALL RDN(U1)
      CALL RDN(U2)
      PI2=6.28318530717958648D0
      X1=DSQRT(-2.D0*ALOG(U1))*DCOS(PI2*U2)
      X2=DSQRT(-2.D0*ALOG(U1))*DSIN(PI2*U2)
      X=X1
   10 RHO=DEXP(-DELT/TAU)
      XCUR=RHO*XLAST+DSQRT(1-RHO**2)*X
      ERR=SIGMA*XCUR
      RETURN
      END
```

The variable X2 needs to be zeroed at the beginning of the simulation.

3.2 INS Error Model

The INS error model introduces errors into the horizontal navigation channels. These errors have the following parameters:

Velocity Error Standard Deviation	= 0.707 n.m./hr. per axis
Velocity Error Mean Error	= 0 per axis
Velocity Error Correlation Time	= 30 min. per axis

Since the INS model does not integrate velocity, the velocity error is integrated to form position errors to be added to INS position outputs. First, two velocity errors V_n and V_E are derived. These velocity errors are integrated to obtain the latitude error and the longitude error ,

$$\delta\lambda = \delta\lambda + (\delta V_n/R_o)\Delta t$$
$$\delta\phi = \delta\phi + (\delta V_e/R_o \cos\lambda)\Delta t$$

where R_o is the radius of the earth. Then the noise-corrupted direction cosines, longitude, and velocity outputs are:

$$C_{XX} = C_{XX} - \cos\alpha \sin\lambda\,\delta\lambda$$
$$C_{XY} = C_{XY} + \sin\alpha \sin\lambda\,\delta\lambda$$
$$C_{XZ} = C_{XZ} + \cos\lambda\, d\lambda$$
$$\phi = \phi + \delta\phi$$
$$V_N = V_N + \delta V_N$$
$$V_E = V_E + \delta V_E$$

where is the wander angle. The following codes are inserted at the end of INSIN and INSNAV, respectively:

<u>Code To Be Inserted In INUIN Subroutine</u>

```
DOUBLE PRECISION XLINVE,XLINVN,DLINVE,DLINVN,PI2
PI2=6.28318530717958648D0
CALL RDN(U1)
CALL RDN(U2)
XLINVE=DSQRT(-2.DO*ALOG(U1))*COS(PI2*U2)
XLINVN=DSQRT(-2.DO*ALOG(U1))*SIN(PI2*U2)
DELLAT=0.
DELLNG=0.
SDINVE=(SQRT(2.)/2.)/(3600.*FPSKTS)
TAUINV=1800.
SDINVN=SDINVE
IF((IERROR.AND."000007).EQ."000001) SDINVN=0.
IF((IERROR.AND."000007).EQ."000004) SDINVN=5*SDINVN
IF((IERROR.AND."000070).EQ."000010) SDINVE=0
IF((IERROR.AND."000070).EQ."000040) SDINVE=5*SDINVE
```

Code To Be Inserted In INUNAV Subroutine

```
      DOUBLE PRECISION XCINVE,XCINVN
      IF((IERROR.AND."000077).EQ."000011) GO TO 100
      CALL RNOISE(XLINVE,DTINUS,TAUINV,SDINVE,DLINVN,XCINVN)
      CALL RNOISE(XLINVN,DTINUS,TAUINV,SDINVN,DLINVE,XCINVE)
      XLINVE=XCINVE
      XLINVN=XCINVN
      VELEFS=VELEFS+DLINVE
      VELNFS=VELNFS+DLINVN
      DELLAT=DELLAT+(DLINVN/EARADF)*DTINUS
      DELLNG=DELLNG+(DLINVE/(EARADF*COS(ANLATR)))*DTINUS
      FLNGOR=FLNGOR+DELLNG
      CXXDIR=CXXDIR-COS(WANDER)*SIN(ANLATR)*DELLAT
      CXYDIR=CXYDIR+SIN(WANDER)*SIN(ANLATR)*DELLAT
      CXZDIR=CXZDIR+COS(ANLATR)*DELLAT
  100 CONTINUE
```

3.3 TACAN Error Model

Random, time-correlated errors, having the following characteristics, are added to the TACAN range and bearing measurements:

Range Bias Error Standard Deviation	= 2000 ft.
Bearing Bias Error Standard Deviation	= 2 deg.
Range and Bearing Correlation Time	= 5 sec.
Range and Bearing Mean Error	= 0

The following code is inserted into the TACAN and CHNTAC subroutines of DAIS.

Code To Be Inserted Into TACAN Subroutine

```
      DOUBLE PRECISION XLTACR,XLTACB,PI2
      CALL RDN(U1)
      CALL RDN(U2)
      XLTACR=DSQRT(-2.D0*ALOG(U1))*DCOS(PI2*U2)
      XLTACB=DSQRT(-2.D0*ALOG(U1))*DSIN(PI2*U2)
```

```
      SDTACR=2000.
      SDTACB=2./DEGPRR
      TAUTAC=5.
      IF((IERROR.AND."000700).EQ."000100) SDTACR=0.
      IF((IERROR.AND."000700).EQ."000400) SDTACR=5.*SDTACR
      IF((IERROR.AND."007000).EQ."001000) SDTACB=0.
      IF((IERROR.AND."007000).EQ."004000) SDTACB=5.*SDTACB
      DTTACS=DELTIM*(8./ITRATE(MODELN))
```

Code To Be Inserted Into CHNTAC Subroutine

```
      DOUBLE PRECISION XCTACR,XCTACB
      IF((IERROR.AND."007700).EQ."001100) GO TO 100
      CALL RNOISE (XLTACR,DTTACS,TAUTAC,SDTACR,DLTACR,XCTACR)
      CALL RNOISE (XLTACB,DTTACS,TAUTAC,SDTACB,DLTACB,XCTACB)
      XLTACR=XCTACR
      XLTACB=XCTACB
      CRANGF=CRANGF+DLTACR
      CAZMTR=CAZMTR+DLTACB
100   CONTINUE
```

3.4 Air Data Error Model

In order to corrupt true air speed (TAS) without disturbing the calculation of altitude, the temperature output from the air data sensor model is perturbed. The sensitivity of the TAS to changes in the temperature is related as follows:

$$\frac{dT_R}{T_R} = 2\frac{d(TAS)}{TAS}$$

where T_R is the temperature in deg.R. A percentage error in the TAS of k%, 1 , is:

$$T_R = T_R(1 + 2k/100)$$

where R is a random, time-correlated variable with the following parameters:

Percentage Error in TAS = 3%, 1σ
Correlation Time = 3 min.

The following code is inserted in the ADCIN and ADC subroutines to achieve the required temperature corruption.

To Be Inserted In ADCIN Subroutine

```
DOUBLE PRECISION XLADCT
CALL RDN(U1)
CALL RDN(U2)
XLADCT=DSQRT(-2.D0*ALOG(U1))*COS(PI2*U2)
SDADCT=3.
TAUADC=180.
IF((IERROR.AND."070000).EQ."010000) SDADCT=0.
IF((IERROR.AND."070000).EQ."040000) SDADCT=5.*SDADCT
DTADCS=DELTIM*(64./ITRATE(MODELN))
```

To Be Inserted Into ADC Subroutine

```
      DOUBLE PRECISION XCADCT
      IF((IERROR.AND."070000).EQ."010000) GO TO 100
      CALL RNOISE(XLADCT,DTADCT,TAUADC,SDADCT,DLADCT,XCADCT)
      XLADCT=XCADCT
      TMALTR=TMALTR*(1.+2.*DLADCT/100.)
100   CONTINUE
```

IV. THE FUNDAMENTAL STRUCTURE OF KALMAN FILTER

The system state is described by:

$$\underline{X}(K+1) = \underline{\Phi}(K+1, K)\,\underline{X}(K) + \underline{B}(K)\,\underline{U}(K) + \underline{G}(K)\,\underline{W}(K)$$

and the measurements are described by

$$\underline{Z}(K) = \underline{H}(K)\,\underline{X}(K) + \underline{V}(K)$$

where,

$\underline{X}(k)$ = n-by-1 state vector at time t_k

$\underline{U}(k)$ = r-by-1 deterministic input vector

$\underline{W}(k)$ = s-by-1 driving noise vector

$\underline{Z}(k)$ = m-by-1 measurement vector

$\underline{V}(k)$ = m-by-1 measurement noise vector

$\underline{\Phi}(k+1,k)$ = n-by-n state transition matrix

$\underline{B}(k)$ = n-by-r deterministic input matrix

$\underline{G}(k)$ = n-by-s noise input matrix

$\underline{H}(k)$ = m-by-n measurement matrix

and, W(k) and V(k) are zero mean white noise sequences with known covariance:

$$E[\underline{W}(j)\,\underline{W}(k)^T] = \begin{cases} \underline{Q}(k) & j = k \\ 0 & j \neq k \end{cases}$$

$$E[\underline{V}(j)\,\underline{V}(k)^T] = \begin{cases} \underline{R}(k) & j = k \\ 0 & j \neq k \end{cases}$$

$$E[\underline{W}(j)\,\underline{V}(k)^T] = \underline{0}$$

The Kalman filter updates the state estimate $\underline{X}(k)$ and error covariance $\underline{P}(k)$ at the measurement time t_k by:

$$\hat{\underline{x}}(k) = \hat{\underline{x}}(k^-) + \underline{K}(k)\,\underline{\nu}(k)$$

$$\underline{P}(k) = \underline{P}(k^-) - \underline{K}(k)\,\underline{H}(k)\,\underline{P}(k^-)$$

From one measurement to the next, the state and error covariance are extrapolated as follows:

$$\hat{\underline{x}}(k^-) = \underline{\Phi}(k,k-1)\,\hat{\underline{x}}(k-1) + \underline{B}(k-1)\,\underline{U}(k-1)$$

$$\underline{P}(k^-) = \underline{\Phi}(k,k-1)\,\underline{P}(k-1)\,\underline{\Phi}(k,k-1)^T + \underline{G}(k-1)\,\underline{Q}(k-1)\,\underline{G}(k-1)^T$$

where K(k) is the Kalman filter gain and Y(k) is the innovation process,

$$\underline{K}(k) = \underline{P}(k^-)\,\underline{H}(k)^T\,\underline{V}(k)^{-1}$$

$$\nu(k) = \underline{Z}(k) - H(k)\,\hat{\underline{x}}(k^-)$$

and

$$\underline{V}(k) = E[V(k)\,V(k)^T]$$
$$= \underline{H}(k)\,\underline{P}(k^-)\,\underline{H}(k)^T + \underline{R}(k)$$

The above recursive relations are initiated by apriori knowledge of the state and the error covariance at time . Thus,

$$\hat{\underline{X}}(0) = \hat{\underline{X}}_0$$
$$\underline{P}(0) = \underline{P}_0$$

V. FULL-SCALE STATE SPACE ERROR MODELS

The comprehensive error models, which are the departing point of the models simplifications, are presented in this section (Ref 1).

5.1 Inertial System Error Model

With the latitude-longitude mechanization (x: level east, y: level north, z: up) of the platform orientation, the full-scale inertial system errors model is developed in terms of the following state variables:

(1) Position errors $(\delta R_e, \delta R_n, \delta R_v)$ defined as the computed values minus the true values,

(2) Velocity errors $(\delta V_e, \delta V_n, \delta V_v)$ defined similarly,

(3) Misalignment angles defined as angles from the computer axis to the platform axes,

(4) Accelerometer errors $(\alpha_e, \alpha_n, \alpha_v)$

(5) Platform drift errors $(\epsilon_e, \epsilon_n, \epsilon_v)$

Because of the limited cross-feed between the vertical loop and the level loop and due to the fact that the vertical loop needs to be stabilized by the barometer altimeter measurement of the altitude, the vertical and the level loops are decoupled.

5.1.1 Inertial System Level Loop State Equations

$$\delta \dot{R}_e = ((V_e/R)\tan L)\delta R_n + \delta V_e$$
$$\delta \dot{R}_n = ((V_e/R)\tan L)\delta R_e + \delta V_n$$
$$\delta \dot{V}_e = -\omega_s^2 \delta R_e + (2W_{ie}\sin L + (V_e/R)\tan L)\delta V_n - a_u \psi_n + a_n \psi_u + \alpha_e + \xi_e$$
$$\delta \dot{V}_n = -\omega_s^2 \delta R_n - (2W_{ie}\sin L + (V_e/R)\tan L)\delta V_e - a_e \psi_n + a_u \psi_e + \xi_n$$
$$\dot{\psi}_e = \omega_u \psi_n - \omega_n \psi_u + \epsilon_e$$
$$\dot{\psi}_n = -\omega_u \psi_e + \omega_e \psi_u + \epsilon_n$$
$$\dot{\psi}_u = \omega_n \psi_e - \omega_e \psi_n + \epsilon_u$$
$$\dot{\alpha}_e = -(1/T_{\alpha e})\alpha_e + (1/T_{\alpha e})\eta_{\alpha e}$$
$$\dot{\alpha}_n = -(1/T_{\alpha n})\alpha_n + (1/T_{\alpha n})\eta_{\alpha n}$$
$$\dot{\epsilon}_e = -(1/T_{\epsilon e})\epsilon_e + (1/T_{\epsilon e})\eta_{\epsilon e}$$
$$\dot{\epsilon}_n = -(1/T_{\epsilon n})\epsilon_n + (1/T_{\epsilon n})\eta_{\epsilon n}$$
$$\dot{\epsilon}_u = -(1/T_{\epsilon u})\epsilon_u + (1/T_{\epsilon u})\eta_{\epsilon u}$$

where

- R = radius of the earth
- L = latitude
- W_s = Schuler frequency
- W_{ie} = earth rotation rate
- a_i = accelerometer outputs, i=e, n, u
- w_i = gyro outputs, i=e, n, u
- T = correlation times
- = white zero-mean Gaussian noises
- = errors in the gravity vector, i=e, n

5.1.2 Baro-Inertial Vertical Loop State Equations

The vertical state equations are:

$$\delta \dot{x}_1 = -c_1 \delta x_1 + (1 + c_1 c_4)\delta x_2 + c_2 \delta h_B$$
$$\delta \dot{x}_2 = (2\omega_s^2 - c_2)\delta x_1 + c_2 c_4 \delta x_2 - \delta x_3 + \delta \eta_B + \delta(a_u - g)$$
$$\delta \dot{x}_3 = c_3 \delta x_1 - c_3 c_4 \delta x_2 - c_3 \delta(a_u - g).$$

where

$$\delta x_1 = \delta R_z$$
$$\delta x_2 = \delta V_z$$
$$\delta x_3 = c_3(\delta R_z - c_4 \delta V_z - \delta h_B)$$

The C_i values are selected to obtain a desired vertical loop time constant.

5.2.1 Measurement Error Model For TACAN (Ref 1)

The bearing and the range error measurements are:

$$\delta\beta = \beta_{INS} - \beta_{TACAN}$$
$$= (y/A)\,\delta R_e - (x/A)\,\delta R_n - b_\beta + \eta_\beta$$

$$\delta\rho = \rho_{INS} - \rho_{TACAN}$$
$$= (x/B)\,\delta R_e + (y/B)\,\delta R_n - b_\rho + \eta_\rho$$

where

$$x = R\cos L_{A/C}\,(l_{A/C} - l_{TACAN})$$
$$y = R(L_{A/C} - L_{TACAN})$$
$$z = h_{A/C} - h_{TACAN}$$
$$A = $$
$$B = x^2 + y^2$$
$$\beta_{TACAN} = (x^2 + y^2 + z^2)^{1/2}$$
$$\beta_{INS} = \beta_{True} + b_\beta - \eta_\beta$$
$$\rho_{TACAN} = \beta_{True} + (\partial\beta/\partial x)_{INS}\,\delta R_e + \left(\frac{\partial\beta}{\partial y}\right)_{INS}\delta R_n$$
$$\rho_{INS} = \rho_{True} + (\partial\rho/\partial x)_{INS}\,\delta R_e + (\partial\rho/\partial y)_{INS}\,\delta R_n$$

The TACAN measurement biases b_β and b_ρ are represented as state variables,

$$\dot{b}_\beta = \eta_{b\beta}$$
$$\dot{b}_\rho = \eta_{b\rho}$$

5.2.2 Measurement Error Model For ILS (Ref 1)

The localizer measurement λ and the glidescope measurement S are modelled as follows:

$$\lambda = \lambda_{true} + \delta\lambda - V_\lambda$$
$$S = S_{true} + \delta S - V_S$$

where V_λ, V_S are zero mean white noise, and $\delta\lambda, \delta S$ are zero mean exponentially correlated variables which will produce two extra state variables. The measurement errors to be used are:

$$\delta z_\lambda = \lambda_{INS} - \lambda_{ILS}$$
$$= (Y/A)\delta R_e - (x/A)\delta R_n - \delta\lambda + V_\lambda$$
$$\delta z_s = S_{INS} - S_{ILS}$$
$$= (\beta/(A+z)^2)\,\delta R_n - \delta S + V_S$$

where

$$x = R\cos L_{A/C}(l_{A/C} - l_{ILS})$$
$$y = R(L_{A/C} - L_{ILS})$$
$$z = h_{A/C} - h_{ILS}$$
$$A = x^2 + y^2$$
$$B = \sqrt{A}$$

5.3 Simplified Models (Ref 1)

A Kalman filter based on the comprehensive models of the previous section is impractical because of the excessive computation burden on the on-board computer. In order to develop a workable Kalman filter, the underlying models must be simplified. These simplifications will result in the filter performance degradation. Some of these simplifications may result in an unacceptable loss in the performance of the resultant filter. *Thus, it is essential that the performance sensitivity* to each simplification be evaluated. The rest of this section outlines some of the feasible simplifications.

5.3.1 Simplified Inertial Level Loop

The step-by-step simplifications are performed in the following order:

Step 1 All the terms in the comprehensive state equations which are of the order of magnitude of w_{ie} are ignored, resulting in removal of the weak coupling between the states. The result will not be the reduction of the number of states, but will make the state transition matrix sparse.

Step 2 Model the accelerometer output uncertainties by a white noise of appropriate power. This will result in reducing the number of states by two.

Step 3 The three states modelling the gyro drift rates are removed by replacing the gyro drift by a white noise of appropriate intensity.

After the above three-step simplification, the state equations become:

$$
\begin{aligned}
\delta \dot{R}_e &= \delta V_e \\
\delta \dot{R}_n &= \delta V_n \\
\delta \dot{V}_e &= -\omega_s^2 \delta R_e - a_u \psi_n + a_n \psi_u + \alpha_e + \xi_e \\
\delta \dot{V}_n &= -\omega_s^2 \delta R_n - a_u \psi_e - a_e \psi_u + \alpha_n + \xi_n \\
\dot{\psi}_e &= \omega_u \psi_n - \omega_n \psi_u + \epsilon_e \\
\dot{\psi}_n &= -\omega_u \psi_e + \omega_e \psi_u + \epsilon_n \\
\dot{\psi}_u &= \omega_n \psi_e - \omega_e \psi_n + \epsilon_u
\end{aligned}
$$

5.3.2 Simplified Vertical Loop

Although the undamped vertical loop is unstable for the short period during approach and landing, the Kalman filter can be developed using the undamped vertical loop dynamics.

$$
\begin{aligned}
\delta \dot{h} &= \delta V_u \\
\delta \dot{V}_u &= \alpha_u + \eta_u
\end{aligned}
$$

where η_u is a white noise representing the term $(-a_n \psi_e + a_e \psi_n)$

5.3.3 Simplified TACAN Model

The state due to the TACAN range bias is removed.

REFERENCES

1. P. S. Mayback, "Filter Design for a TACAN-Aided Baro-Inertial System with ILS Smoothing Capability," AFFDL-TM-7U-52, May 1974.

2. W. G. Heller, "Free Inertial and Damped Inertial Navigation Mechanization and Error Equations," The Analytic Science Corp., Report No. TR-312-2, October 1974.

3. W. G. Heller, "Covariance Propagation Equations for Optimal and Suboptimal Kalman Filter for Integrated Multisensor Inertial Systems," The Analytic Science Corp., Report No. TR-312-1, July 1974.

4. W. G. Heller, "Models for Aided Inertial Navigation System Sensor Errors," The Analytic Science Corp., Report No. TR-312-3, February 1975.

5. S. H. Musick, "A Computer Program for Generating Flight Profile," AFAL-TR-76-247, 1976.

6. E. L. Hamilton, G. Chitwood, and R. M. Revees, "The General Covariance Analysis Program (GCAP), An Efficient Implementation of the Covariance Analysis Equations," Air Force Avionics Laboratory, WPAFB, AFAL-TR-76-247, June 1976.

7. S. H. Musick, "SOFE: A Generalized Digital Simulation for Optimal Filter Evaluation User's Manual," AFWAL-TR-80-1108, October 1980.

8. R. E. Feldman, S. H. Musick, "SOFEPL: A Plotting Postprocessor for 'SOFE', User's Manual," AFWAL-TR-80-1109, to be published.

9. J. C. Pinson, "Inertial Guidance for Cruise Vehicles," in C. T. Leondes Ed., Guidance and Control of Aerospace Vehicles, McGraw-Hill Book Company, New York, 1963.

10. R. L. Blanchard, "A New Algorithm for Computing Inertial Altitude and Vertical Velocity," IEEE Transactions on Aerospace and Electronic Systems, Vol AES-7, No. 6, November 1971.

END

FILMED

9-84

DTI